分布式入侵检测中基于能力与负载的数据分割算法 *

张润莲 [a, b]，李　豪 [a]，叶志博 [a]，赵新红 [c]

(桂林电子科技大学 a. 广西可信软件重点实验室; b. 广西无线宽带通信与信号处理重点实验室; c. 广西密码学与信息安全重点实验室，广西 桂林 541004)

摘　要：针对高速网络环境下分布式入侵检测中海量数据并行检测处理的效率和检测率问题，提出一种基于能力与负载的数据分割算法。该算法依据采集到的集群内各数据分析节点的系统性能指标及运行状态，评估节点的数据处理能力与负载程度。基于节点的能力与负载适应因子，权衡节点在集群中检测和分析数据能力的权重，实现海量数据在集群内各数据分析节点间的动态数据分割，为节点分配适应其能力与实时负载的数据粒度。仿真测试结果表明，该算法具有较好的负载均衡性，降低了系统的检测时间，提高了数据并行处理的效率和检测率。

关键词：分布式入侵检测；数据分割；数据分配；负载均衡

中图分类号：TP393　　doi: 10.3969/j.issn.1001-3695.2017.08.0725

Data partitioning algorithm based on capacity and workload in DIDS

Zhang Runlian[a, b], Li Hao[a], Ye Zhibo[a], Zhao Xinhong[c]

(*a. Guangxi Key Laboratory of Trusted Software*, *b. Guangxi Wireless Broadband Communication & Signal Processing Key Laboratory*, *c. Guangxi Key Laboratory of Cryptography & Information Security*, *Guilin University of Electronic Technology*, *Guilin Guangxi* 541004, *China*)

Abstract: Aiming at the efficiency and detection rate problems of the massive data parallel detecting in high speed network distributed intrusion detection, this paper proposed a data partitioning algorithm based on capacity and workload. In this algorithm, according to the collected performance indicators and running status of data analysis nodes for parallel data processing in the cluster, it evaluated the data processing capacity and the workload of nodes. Based on the node's capacity and load adjustment factor, realized the dynamic data partition among the data analysis nodes by considering the weight of the node for detecting and analyzing data in the cluster. It made the partitioned data granularity of the node matches with node 's capacity and real-time load. The tested results show that the proposed algorithm can reduce the detection time, improve the efficiency and detection rate of the data parallel processing.

Key Words: distributed instruction detection; data partitioning; data allocation; load balance

0　引言

数据的分布采集和并行检测处理，是分布式入侵检测系统(distributed instruction detection system, DIDS)能够处理高速网络中海量数据的关键。并行数据的粒度大小，即分配给节点进行并行处理的数据子集，影响系统的吞吐量和负载均衡。数据分割算法则是按照特定的方式，将海量数据分割成不同子集的方法。数据分割算法的好坏，决定了数据分配的合理性，也影响着系统负载均衡和系统性能。

入侵检测技术作为一种主动的网络安全保护技术，已成为网络安全防御系统的重要组成部分。为提高系统的检测效率，针对基于检测规则的检测方法，文献[1]提出对字符串匹配算法的改进，增加检测系统的吞吐率；针对异常检测方法，文献[2]采用 SVM 及分类方法提高检测速度。为提高检测率，文献[3]通过提高数据包的抓取、过滤和匹配速度，降低在高速网络环境下的丢包率，提高检测的准确率。为提高系统检测的自适应性和扩展性，文献[4]基于 GHSOM 神经网络对新出现的攻击进行增量式学习。针对海量数据的检测处理，文献[5]采用数据分割方法，提出数据流的分片机制；文献[6]将大数据集随机分割后分发给独立的神经网络并行学习；文献[7]采用一种分布式离

基金项目：国家自然科学基金资助项目（61572148）；广西可信软件重点实验室基金（kx201622）；广西无线宽带通信与信号处理重点实验室主任基金项目（GXKL061510，GXKL0614110）；广西密码学与信息安全重点实验室（GCIS201623）；研究生创新项目（2017YJCX26）

作者简介：张润莲（1974-），女，山西介休人，副教授，博士，主要研究方向为信息安全、分布式计算（zhangrl@guet.edu.cn）；李豪（1990-），男，山西临汾人，硕士研究生，主要研究方向为信息安全；叶志博（1994-），男，河南汝州人，硕士研究生，主要研究方向为信息安全；赵新红（1993-），女，河南周口人，硕士研究生，主要研究方向为信息安全.

群点检测算法，均衡各计算节点的工作负载。

数据分割算法被广泛应用在并行计算和大规模数据处理中，维持负载均衡，提高吞吐量。文献[8]通过构造分割函数及算法，实现数据流分割和并行处理；文献[9]按照多轮分配分区策略调整对 Reducer 的数据指派，解决 MapReduce 中数据划分引起的数据倾斜问题；文献[10]针对数据密集型应用，采集并评估节点性能，调整数据分配的大小与数量，提高效率。

本文针对 DIDS 中海量数据的并行检测处理问题，提出一种基于能力与负载的数据分割算法，通过监控、采集集群内各节点的性能指标和实时运行状态，评估节点的数据处理能力和负载；权衡各节点能力与负载在集群节点中的比重，并以之进行数据分割和数据分配，有效发挥节点的能力，维持系统负载均衡，提高系统效率和检测率。

1 分布式入侵检测系统结构

在分布式入侵检测中，为提高系统性能，通过多个分布的传感器节点并行采集网络数据，再将采集的海量数据分割并分发到多个节点并行检测处理，提高系统的数据处理能力。本文采用一种基于能力和负载的数据分割算法进行数据分割，系统结构如图 1 所示，包括数据采集、控制中心和数据分析检测三部分，控制中心包括系统监控、任务调度器和报警响应。

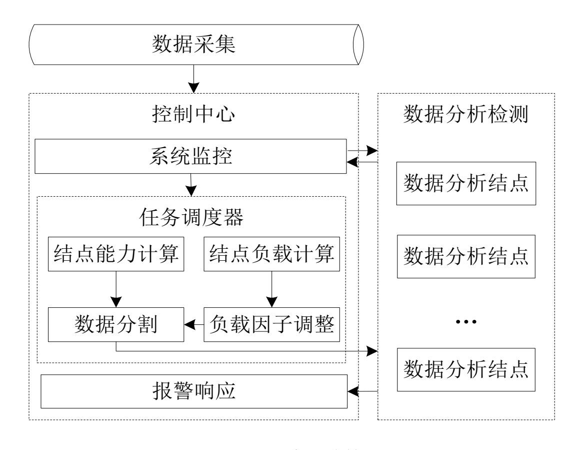


图 1 系统结构

图 1 中，在控制中心，系统监控采集集群内各数据分析节点的性能指标及运行情况，包括 CPU/内存/硬盘大小和利用率等，为数据分割提供决策支持。任务调度器基于数据分割算法，评估节点的能力和负载，进行数据分割。报警响应对各数据分析节点的入侵检测结果进行报警响应。

数据分析检测由各异构节点对采集及分配的数据检测，将检测结果反馈给报警响应模块处理。

2 基于能力与负载的数据分割算法

本文通过对节点能力和负载的评估，为节点动态分割数据子集。

2.1 节点能力计算

节点的能力是决定分配给节点数据粒度大小的重要依据，其主要由节点的硬件配置确定，通过系统监控可以采集相关性能指标。本文对节点的数据处理能力主要考虑节点 CPU、内存和网络带宽的大小，以 $CAP(i)$表示 i 节点的数据处理能力，计算公式如下：

$$CAP(i) = a_1 \times CPU_i + a_2 \times M_i + a_3 \times N_i \quad (1)$$

其中：a_1、a_2、a_3 为节点 CPU、内存和网络带宽对数据处理能力的影响因子，$a_1+a_2+a_3=1$；CPU_i、M_i、N_i 分别为 i 节点的 CPU 个数与 MIPS 大小的乘积、内存大小和网络带宽。因不同指标数据单位不一、数值取值范围可能相差较大，在计算前先对采集的指标数据采用 min-max 线性函数转换方法进行归一化处理，使得数据能够较好地收敛到一个相同的区间。式(1)中的 CPU_i、M_i、N_i 均为归一化后的结果。

节点的异构性，使得在数据分割和分配时缺乏对比和参照的基准，数据粒度的大小划分也难以控制。针对该问题，基于节点的数据处理能力，构造一个反映节点间能力对比参照的指标——节点能力因子，以 $CL(i)$表示 i 节点的能力因子，计算方法如下：

$$CL(i) = \frac{CAP(i)}{Min\{CAP(j)\,|\,1 \le j \le m\}} \quad (2)$$

其中：m 为集群中节点的数量。

2.2 节点负载计算

节点的负载是影响分配给节点数据粒度大小的另一个重要因素。对数据分析节点的负载评估，根据系统监控采集的节点 CPU、内存和网络带宽的利用率完成。以 $L(i)$表示 i 节点的负载，计算公式如下：

$$L(i) = b_1 \times L(CPU_i) + b_2 \times L(M_i) + b_3 \times L(N_i) \quad (3)$$

其中：$L(CPU_i)$、$L(M_i)$、$L(N_i)$分别为 i 节点的 CPU、内存和网络带宽的利用率；b_1、b_2、b_3 为 i 节点 CPU、内存和网络带宽的利用率对节点负载的影响因子，$b_1+b_2+b_3=1$。

2.3 数据分割及分配

为保证数据分割的合理性，在数据分配时，先判断节点的负载状态，避免为已过载节点分配数据而造成延迟增加。为评估节点负载状态，设置一个负载轻载阈值 α 和一个负载过载阈值 β；并基于节点负载状态构造一个能够反映节点负载对数据分割的数据粒度大小的影响因子——负载适应因子，以 $E(i)$表示 i 节点的负载适应因子，计算公式如下：

$$E(i) = \begin{cases} 1 + \dfrac{|L(i) - \alpha|}{L(i)}, & L(i) \le \alpha \\ 0, & L(i) \ge \beta \\ \dfrac{|L(i) - \beta|}{L(i)}, & \alpha < L(i) < \beta \end{cases} \quad (4)$$

基于对节点能力和负载的评估，以 $W(i)$表示节点的能力与负载在集群内所有数据分析节点中的权重：

$$W(i) = \frac{CL(i) \times E(i)}{\sum_{i=1}^{n} CL(i) \times E(i)} \quad (5)$$

其中：n 为集群内数据分析节点的数量；$CL(i)$、$E(i)$分别为 i 节点能力因子和负载影响因子。在式(5)中，节点的能力越强，负载越轻，其所占权重越大。

以 $D(i)$表示从待处理的海量数据集 M 中，依据节点的权值 $W(i)$分割并分配给 i 节点的数据子集即数据粒度大小，计算公式如下：

$$D(i) = M \times W(i) \tag{6}$$

3 实验结果与分析

3.1 实验环境

本文基于 Hadoop 完成分布式入侵检测系统模型架构，采用 Java 语言实现基于能力与负载的数据分割算法，并将该算法用于控制中心的任务调度，完成数据分割和数据分配。控制中心作为 master 节点，采用 MapReduce 与 slave 节点通信，slave 节点包括系统监控节点和数据分析节点等。系统依托校园网，实现各功能节点的分布部署。其中，Hadoop 集群中的各节点均安装 Ubuntu 10.04 操作系统；Master 配置为 Intel 酷睿 3.3 GHz、4 GB 内存、500 GB 硬盘；系统监控节点、报警响应节点和数据存储节点配置均为 Intel 奔腾 2.8 GHz、2 GB 内存、250 GB 硬盘；各数据分析节点异构。

本文侧重考虑通过对海量数据的并行检测处理，以提高系统性能，故在测试中，主要测试数据分割算法对海量数据分割的合理性，数据分割后分配到数据分析节点并行检测处理时的系统负载均衡性，以及对系统性能的改善情况。本文测试采用经典的 DARPA2000 数据集，下载了约 3 GB 的 DARPA2000 数据集存放在数据存储节点上；在数据分析节点上，为保证检测结果的一致性和有效性，安装配置 Snort 系统，在接收到由 master 分割并分配的待检测数据后，由 Snort 系统进行数据检测；系统监控采用开源工具 Sigar 采集各数据分析节点运行状态；报警响应节点对检测的入侵行为进行报警处理。

基于上述实验环境和实验数据，本节对基于能力与负载的数据分割算法的负载均衡性、检测时间和检测率进行了测试，并与文献[10]的动态负载调整方法、一致性哈希算法[11]进行对比。

本文算法中的相关参数设置如下：

a) 在式(1)中，根据节点 CPU、内存和带宽对数据处理的影响以及不同参数的处理效果，取 a_1=0.4、a_2=0.3、a_3=0.3；

b) 在式(3)中，b_1=0.4、b_2=0.3、b_3=0.3；

c) 根据对系统轻载和过载的经验认知，设置负载轻载阈值 α=0.3，负载过载阈值 β=0.7；

d) 数据分析节点组包含 8 个异构节点。

3.2 实验结果与分析

实验 1 负载均衡测试

本文测试共使用八台性能异构的节点进行数据分析检测，各节点编号依次为 S1、S2、⋯、S8。在仿真测试前，测得各节点的性能比例为：1:3.4:1.4:1.3:1:1.3:1.4:1.1。实验中，分别测试了采用本文算法、文献[10]方法和一致性哈希算法对 3 GB 数据集进行分割并分配到八个节点上的数据子集结果如表 1 所示。

表 1 采用不同算法进行数据分割后的数据量分布结果

节点	算 法		
	一致性哈希算法	文献[10]方法	本文算法
S1	376.8	301.3	291.5
S2	383.3	765.7	783.2
S3	391.1	313.4	302.7
S4	385.3	251.2	244.5
S5	388.1	325.2	343.8
S6	390.5	481.3	490.2
S7	384.4	315.4	296.3
S8	377.7	310.7	319.9

表 1 数据显示，一致性哈希算法通过虚拟节点的映射关系进行数据分割，没有考虑节点的能力与负载，分割和分配到节点的数据子集相差不大；文献[10]方法采用其公式计算节点对数据集的计算时间，并以其大小动态调整分配的数据量，该方法分配给节点的数据量能够适应节点的性能；本文算法在对节点能力进行评估外，进一步考虑了节点的实时负载变化，其分配给节点的数据子集更好地适应了节点的需求。

在上述三种算法将数据分割并分配到八个节点后，采用负载均衡度 $bLBM(t)$[12]对比测试在八个节点上的负载均衡性，测试结果如图 2 所示。

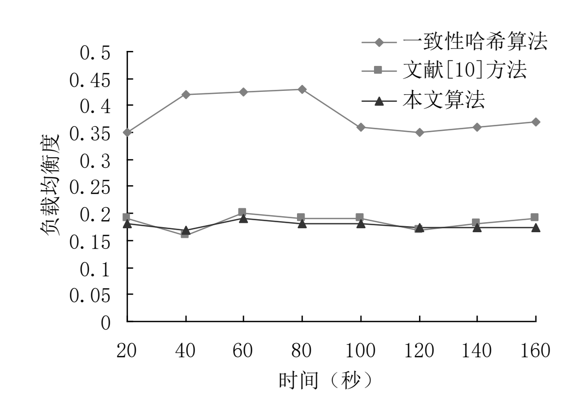


图 2 不同算法负载均衡度对比测试

图 2 显示，一致性哈希算法的 $bLBM(t)$值最大，其因为缺乏负载调整机制，给各节点分配的数据比较均匀，低性能和负载重的节点被分配过多的数据，负载不均衡；本文算法与文献[10]的 $bLBM(t)$值相差较小，但其变化波动更小，系统更加稳定。

实验 2 检测时间测试

节点负载的失衡，将会导致节点响应延迟，影响系统整体性能。基于上述的数据分割和数据分配，分别测试了各节点对被分配的数据子集进行入侵检测的时间开销，结果如表 2 所示。表 2 中，ST 行中的数据为各算法测得的所有节点完成数据检测的系统检测时间总和。

表 2　对不同算法分割数据后的检测时间测试结果

节点	算　法		
	一致性哈希算法	文献[10]方法	本文算法
S1	136.3	77	78
S2	53	76	83
S3	76	70	63
S4	87	75	71
S5	115	90	86
S6	83	97	94
S7	77	68	65
S8	102	86	89
ST	729.3	639	629

表 2 显示，在同样的测试条件下，一致性哈希算法因数据分割和分配的不均衡，其检测任务的时间最长，ST 最大。文献[10]方法与本文算法都进行了负载均衡调整，数据分配较合理，节点检测时间都大大降低，本文算法因更好的负载均衡性，其检测时间稍低。

实验 3　检测率测试

基于上述进行的数据分割和数据分配，测试各节点对被分配数据子集进行入侵检测的检测率。测试结果表明，一致性哈希算法分割后的检测率最低约为 87.3%，主要在于其通过数据包和虚拟节点的映射关系进行数据分割，分割粒度小，对会话的完整性破坏大，降低了检测率。文献[10]方法与本文算法都未考虑数据完整性处理，两者因分割的次数少，对数据的完整性破坏小，两者检测率相近约为 99%。

4　结束语

针对分布式入侵检测中海量数据的并行检测处理问题，提出一种基于能力与负载的数据分割算法，该算法通过评估节点的能力和实际负载情况，在集群内权衡各节点的实际状态及数据分配关系，为数据处理能力强且负载轻的节点分配更多的待处理数据；该算法以节点的实际性能和负载进行数据分割和分配，充分发挥节点的能力，并避免了数据处理能力弱或负载过重的节点被分配大量的数据而影响系统性能。仿真测试结果表明，本文算法对节点的数据分割和分配符合节点的能力与实时负载状态，系统负载均衡性好；而良好的负载均衡，有效降低了系统并行处理的时间开销，提高了系统性能和检测率。在下一步的工作中，将考虑如何在数据分割中保证数据的完整性，进一步提高检测率。

参考文献:

[1] Choi Y H, Jung M Y, Seo S W. A fast pattern matching algorithm with multi-byte search unit for high-speed network security [J]. Computer Communications, 2011, 34 (14): 1750-1763.

[2] Erfani S M, Rajasegarar S, Karunasekera S, et al. High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning [J]. Pattern Recognition, 2016, 58: 121-134.

[3] Song H, Lockwood J W. Efficient packet classification for network intrusion detection using FPGA [C]// Proc of the 13th International Symposium on Field-Programmable Gate Arrays. New York: ACM Press, 2005: 238-245.

[4] 杨雅辉, 黄海珍, 沈晴霓, 等. 基于增量式 GHSOM 神经网络模型的入侵检测研究 [J]. 计算机学报, 2014, 37 (5): 1216-1224.

[5] Kruegel C, Valeur F, Vigna G, et al. Stateful intrusion detection for high-speed networks [C]// Proc of IEEE Symposium on Security and Privacy. Washington: IEEE Computer Society, 2002: 285-294.

[6] 刘衍珩, 田大新, 余雪岗, 等. 基于分布式学习的大规模网络入侵检测算法 [J]. 软件学报, 2008, 19 (4): 993-1003.

[7] 王习特, 申德荣, 白梅, 等. BOD: 一种高效的分布式离群点检测算法 [J]. 计算机学报, 2016, 39 (1): 36-51.

[8] Gedik B. Partitioning functions for stateful data parallelism in stream processing [J]. The VLDB Journal, 2014, 23 (4): 517-539.

[9] 王卓, 陈群, 李战怀, 等. 基于增量式分区策略的 MapReduce 数据均衡方法 [J]. 计算机学报, 2016, 39 (1): 19-35.

[10] Rosas C, Sikora A, Jorba J. Dynamic tuning of the workload partition factor and the resource utilization in data-intensive applications [J]. Future Generation Computer Systems, 2014, 37 (6): 162-177.

[11] Karger D R, Lehman E, Leighton T, et al. Consistent hashing and random trees: distributed caching protocols for relieving hot spots on the world wide Web [C]// Proc of International Symposium on Theory of Computing. 1997: 654–663.

[12] 陈一骄, 卢锡城, 时向泉, 等. 一种面向会话的自适应负载均衡算法 [J]. 软件学报, 2008, 19 (7): 1828-1836.